Cost-Optimal Planning with Landmarks

Erez Karpas and Carmel Domshlak
Faculty of Industrial Engineering & Management
Technion, Israel

Filip-Martin Brinkmann
filip.brinkmann@unibas.ch

December 13, 2012

Overview

Landmarks recapitulation

Landmarks in the SAS^+ formalism

A SAS^+ task is a 5-tuple $\Pi = \langle V, A, s_0, G, C \rangle$ with

- $V = \{v_1, ..., v_n\}$: set of *state variables*. Each v_i is associated with a finite domain $dom(v_i)$.
- $F := \bigcup_i dom(v_i)$: set of *facts*, assuming name uniqueness.
- A : set of *actions*. An action is a pair $\langle pre(a), eff(a) \rangle$ of partial assignments to V.
- C : Each action has a non-negative cost $C(a)$.
- s_0 : initial state.
- G : Goal - partial assignment to V.

Landmarks recapitulation

Landmarks

Be ϕ be a formula over facts F.
It is called a *landmark* if it is true at some time in any plan for Π.
In the following, we only treat landmarks that are *disjunctions of facts*. A set of landmarks is denoted by L.

Landmark Cost Estimates

Landmark Count Heuristic

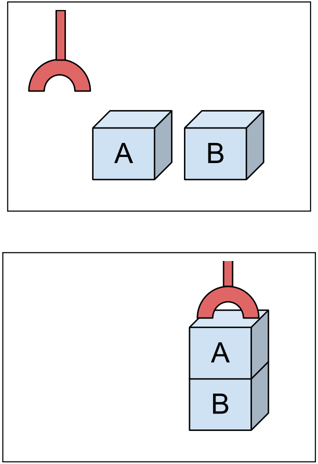
$L(s, \pi) := L \setminus (Accepted(s, \pi) \setminus ReqAgain(s, \pi))$
(Set of LMs to be reached from s)

- $s := state$
- $\pi := \langle a_1, ..., a_k \rangle$: sequence of actions, starting at s_0

$h(s, \pi) := |L(s, \pi)|$: cost estimate

- The LM heuristic $|L(s, \pi)|$ can overestimate and is thus not admissible.
- Why is that so?

Landmark Cost Estimates Are Not Admissible



Example: Blocksworld Task

- $L(s, \pi) = \{crane\text{-}empty,\ on(A,B)\}$
- $\rightarrow |L(s, \pi)| = 2$
- However: only one action *stack(A,B)* does the job
- $\rightarrow |L(s, \pi)|$ is not admissible

1 Introduction

2 Admissible Landmark Heuristics

3 Admissible Heuristics with Action Landmarks

4 Evaluations
- Contributions: ALC and LM-A*
- Competitivity of LM-A*

General Idea

What needs to be done to make LM heuristics admissible?
Distribute action costs over all LMs and maintain upper bound.

Limiting Action Costs

Action-to-Landmark costs

Given

- ϕ : *landmark*
- $C(a)$: action cost of a
- $cost(a, \phi)$: cost of a assigned to ϕ

Then

$$\forall a \in A : \sum_{\phi \in L(a|s,\pi)} cost(a, \phi) \leq C(a) \tag{1}$$

This shares the overall cost of a between all LMs on path π that a applies to.

Minimizing Landmark Costs

Landmark costs

$$\forall \phi \in L(s,\pi) : cost(\phi) \leq min_{a \in ach(\phi|s,\pi)} cost(a,\phi) \quad (2)$$

with

- $ach(\phi \mid s,\pi) \subseteq A$: subset of actions which lead to ϕ on their way π to the goal: achievers
- $L(a \mid s,\pi) = \{\phi \mid \phi \in L(s,\pi), a \in ach(\phi|s,\pi)\}$: Landmarks achievable by a on path π

The costs for each landmark are thus less or equal than the minimum cost assigned to ϕ by all actions achieving it.

Admissible Heuristic Estimate

This gives us an admissible estimate of the goal distance $h^*(s)$.

Proposition 1

Assuming we have a set of action-to-landmark and landmark costs satisfying (1,2). Then

$$h_L(s, \pi) := cost(L(s, \pi)) := \sum_{\phi \in L(s,\pi)} cost(\phi)$$

is an *admissible estimate* of $h^*(s)$.

However, we have not decided on *how to partition costs* among actions.

Cost Partitioning: Overview

The simplest way to partition costs is by uniform cost sharing.

- Fast and easy.
- However, this may lead to sub-optimal sharing.

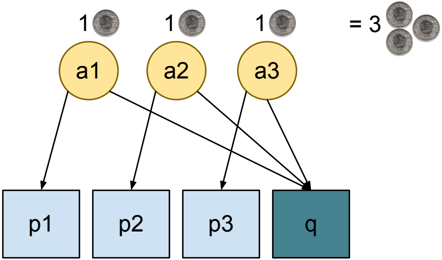
There is also an optimal cost sharing.

- Achievable via linear programming.
- Can be computed in polynomial time.

Let us have a closer look...

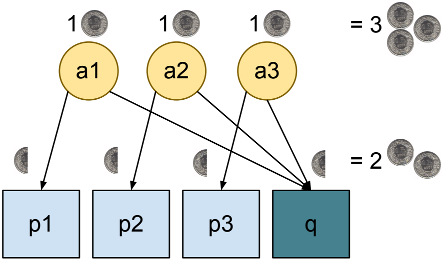
Cost Partitioning

- Given 3 actions: $\{a_1, a_2, a_3\}$ and 4 landmarks: $\{p_1, p_2, p_3, q\}$.
- Each action a_i achieves p_i and q.



Cost Partitioning

- Given 3 actions: $\{a_1, a_2, a_3\}$ and 4 landmarks: $\{p_1, p_2, p_3, q\}$.
- Each action a_i achieves p_i and q.

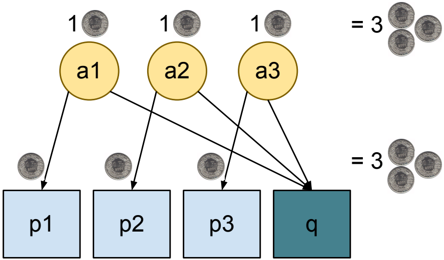


Uniform Cost Sharing

- $cost(a_i, p_i) = cost(a_i, q) = 0.5$
- $\rightarrow cost(p_i) = cost(q) = 0.5$.
- $\rightarrow h_L(s, \pi) = \frac{3}{2} + 0.5$

Cost Partitioning

- Given 3 actions: $\{a_1, a_2, a_3\}$ and 4 landmarks: $\{p_1, p_2, p_3, q\}$.
- Each action a_i achieves p_i and q.



Optimal Cost Sharing

- $cost(a_i, p_i) = 1$
  $\rightarrow cost(p_i) = 1$
- $cost(a_i, q) = 0$
  $\rightarrow cost(q) = 0$
- $\rightarrow h_L(s, \pi) = 3$

1 Introduction

2 Admissible Landmark Heuristics

3 Admissible Heuristics with Action Landmarks

4 Evaluations
- Contributions: ALC and LM-A*
- Competitivity of LM-A*

Further Improvement with Action Landmarks

So far, we considered only *fact landmarks*.
However, there also exist action landmarks [Zhu and Givan, 2004; Vidal and Geffner, 2006].
Action landmarks

- are actions that will be performed on every path of the planning task (at least once).
- If no plan without the action exists, it is an action landmark.

Heuristic Estimate with Action Landmarks

1. Sum up all costs of unused action landmarks $U(s, \pi)$.
2. Remove all LMs achievable by $U(s, \pi)$ from $L(s, \pi)$.
3. Perform action cost sharing over remaining landmarks.

This leads to improved heuristic estimate h_{LA}:

Action Landmark Covering

$$h_{LA} := cost(L_U(s, \pi)) + \sum_{a \in U(s,\pi)} C(a) \quad (3)$$

with

Heuristic Estimate with Action Landmarks

1. Sum up all costs of unused action landmarks $U(s, \pi)$.
2. Remove all LMs achievable by $U(s, \pi)$ from $L(s, \pi)$.
3. Perform action cost sharing over remaining landmarks.

This leads to improved heuristic estimate h_{LA}:

Action Landmark Covering

$$h_{LA} := cost(L_U(s, \pi)) + \sum_{a \in U(s,\pi)} C(a) \quad (3)$$

with

- $L_U(s, \pi) = L(s, \pi) \setminus \bigcup_{a \in U(s,\pi)} L(a \mid s, \pi)$

Heuristic Estimate with Action Landmarks

1. Sum up all costs of unused action landmarks $U(s,\pi)$.
2. Remove all LMs achievable by $U(s,\pi)$ from $L(s,\pi)$.
3. Perform action cost sharing over remaining landmarks.

This leads to improved heuristic estimate h_{LA}:

Action Landmark Covering

$$h_{LA} := cost(L_U(s,\pi)) + \sum_{a \in U(s,\pi)} C(a) \tag{3}$$

with

- $L_U(s,\pi) = L(s,\pi) \setminus \bigcup_{a \in U(s,\pi)} L(a \mid s,\pi)$
- $cost(L_U(s,\pi))$ computed via action cost sharing

1 Introduction

2 Admissible Landmark Heuristics

3 Admissible Heuristics with Action Landmarks

4 Evaluations
- Contributions: ALC and LM-A*
- Competitivity of LM-A*

Setup

- Infrastructure:
    - Fast Downward Planner
    - Landmark discovery of LAMA
- Limits: 30 mins, 1.5 GB of memory
- Hardware: 3GHz Intel E8400 CPU

Evaluations Overview

Two different angles are used:

1. Contributions of action landmark covering and LM-A*.
   - h_{LA} vs. h_L
   - A* vs. LM-A*
2. Competitivity of LM-A*/h_{LA} vs. forward-search cost-optimal planning.

1. Introduction
2. Admissible Landmark Heuristics
3. Admissible Heuristics with Action Landmarks
4. Evaluations
   - Contributions: ALC and LM-A*
   - Competitivity of LM-A*

Contributions of Action Landmark Covering and LM-A*

- A* vs. LM-A*
  - both with h_{LA}
  - $LM - A^*$ solves
    - 15% more tasks in Blocksworld,
    - 43% more tasks in Depots,
    - 53% more tasks in Logistics and
    - 0% more tasks in Satellite
- h_L vs. h_{LA}
  - both $LM - A^*$
  - h_{LA} solving two more tasks than h_L
  - savings in time and expanded nodes up to 70%

1. Introduction
2. Admissible Landmark Heuristics
3. Admissible Heuristics with Action Landmarks
4. Evaluations
   - Contributions: ALC and LM-A*
   - Competitivity of LM-A*

LM-A* / h_{LA} vs. M&S : Setup

- Baseline evaluations against A^* with h_{max} and *blind search*.
- State-of-the-art forward search: merge-and-shrink (M&S).
- M&S is used with 10^4 and 10^5 fixed abstraction bounds.
- Both heuristics implemented in same planning system.

LM-A* / h_{LA} vs. M&S : Results

On average, h_{LA}

- solved more tasks,
- expanded less nodes and
- needed less time.

planner	*solved*	*nodes*	*time (sec)*
h_{LA}	273	107,701	44
M&S	251	457,336	101

Discussion

Questions?

Achieving Optimal Cost Partitioning

- Can be computed in polynomial time.
- Form linear constraints from Eq. (1) and (2).
- Solve linear program for objective $\sum_{\phi \in L(s,\pi)} cost(\phi)$.

Landmark Set Monotonicity

For two landmark sets L and L', with $L \subseteq L'$, the *LP-based cost sharing* assures

$$cost(L') \geq cost(L)$$

This allows for separating landmark discovery and exploitation.
(Or, as the authors put it: *"More landmarks can never hurt."*)

Detailed Evaluation Results I

		$A^* + h_{LA}$		*LM*-$A^* + h_L$		*LM*-$A^* + h_{LA}$		*FA*-10^4		*FA*-10^5		h_{max}		blind	
task	C^*	*time*	*nodes*	*time*	*nodes*	*time*	*nodes*	*time*	*nodes*	*time*	*nodes*	*time*	*nodes*	*time*	*nodes*
	BLOCKSWORLD														
4-0	6	0	15	0	15	0	15	0.03	7	0.03	7	0	25	0.01	95
4-1	10	0.01	13	0	15	0	13	0.04	11	0.03	11	0	23	0.01	66
4-2	6	0.01	7	0.01	8	0	7	0.04	7	0.03	7	0.01	18	0	61
5-0	12	0	31	0.01	41	0	31	0.3	13	0.97	13	0.01	145	0	467
5-1	10	0.01	35	0	45	0	31	0.3	11	0.97	11	0.01	135	0.01	561
5-2	16	0.02	87	0.01	158	0.01	87	0.3	17	0.97	17	0.01	297	0	798
6-0	12	0.01	22	0.01	29	0	22	0.84	13	8.03	13	0.01	276	0.01	1826
6-1	10	0.02	39	0.01	44	0.01	39	0.86	11	8.12	11	0.01	755	0.02	4911
6-2	20	0.15	1146	0.12	943	0.21	933	0.97	733	8.66	68	0.04	2556	0.02	6409
7-0	20	0.06	249	0.06	292	0.08	240	1.92	579	23.04	144	0.11	5943	0.14	36333
7-1	22	1.68	10952	2.09	12224	2.44	9032	1.62	9977	19.47	1887	0.45	33194	0.21	63376
7-2	20	0.38	1941	0.45	2452	0.41	1453	1.56	1855	19.06	611	0.29	18293	0.19	55218
8-0	18	0.22	894	0.41	1762	0.29	827	3.93	5570	38.48	692	2	94671	2.18	519107
8-1	20	1.84	8283	2.63	10810	1.92	5647	4.17	45706	34.82	11880	3.83	199901	2.53	636138
8-2	16	0.03	95	0.12	504	0.08	219	3.55	293	33.56	63	1.28	52717	1.89	434664
9-0	30			175.36	469633	132.87	260857	14.38	1232639	66.73	971416	81.21	3840589	36.58	7983389
9-1	28	6.64	18609	8	22337	7.93	14852	9.41	94991	61.27	58867	30.89	1200345	29.49	5922420
9-2	26	1.63	4712	4.87	12572	3.4	6581	11.19	161653	103.8	20108	31.33	1211463	29.93	5984400
10-0	34							266.2	32869439						
10-1	32					919.68	1393515	201.66	23517042	275.71	12063661				
10-2	34							271.61	33331658	288.23	18457528				
11-0	32							195.71	16219694	138.77	7077045				
12-1	34			612.02	901023	250.78	272199								
	DEPOTS														
1	10	0.01	33	0.01	31	0.02	31	0.01	11	0.01	11	0.01	136	0	329
2	15	0.13	668	0.2	790	0.21	614	2.83	875	1.15	16	0.18	3771	0.11	15404
3	27	193.32	810127	121.36	169765	52.43	72979	24.74	348300	235.01	239255	96.47	1204646	27.16	2930398
4	30					1061.04	1041194	50.08	1284029	458.39	1219026				
7	21	213.66	890532	69.15	86052	29.72	39348	39.12	212544			163.48	1331701	71.79	6501100
10	24			400.99	447195	193.32	199412	156.91	3240433						
13	25			45.07	42808	41.66	27977	122.03	1427824	499.05	1183545				
	LOGISTICS														
4-0	20	0.09	258	0.01	76	0.04	76	0.04	21	0.05	21	0.04	4884	0.06	11246
4-1	19	0.18	1238	0.02	195	0.08	195	0.04	20	0.05	20	0.03	4185	0.05	9249
4-2	15	0.02	111	0.01	53	0.01	53	0.04	16	0.05	16	0.01	1205	0.03	4955
5-0	27	13.19	123081	0.13	936	0.64	936	0.1	28	0.37	28	0.58	74694	0.6	109525
5-1	17	0.16	1012	0.02	181	0.08	181	0.1	18	0.39	18	0.06	6199	0.12	22307
5-2	8	0	9	0.01	9	0.01	9	0.1	9	0.38	9	0.01	280	0	1031
6-0	25	35.49	244606	0.14	934	0.67	934	0.18	26	1.22	26	1.88	202229	3.27	490207

Detailed Evaluation Results II

domain	N^+/N^1	*solved*		*nodes*		*time*	
		h_{LA}	*FA*	h_{LA}	*FA*	h_{LA}	*FA*
airport	18/24	24	18	1395	528152	8	123
blocks	19/23	20	22	89179	1319533	56	13
depots	7/7	7	7	197365	930573	196	56
driverlog	13/14	14	13	109611	765930	53	14
freecell	5/7	7	5	8487	1406793	10	232
grid	2/2	2	2	15313	1705511	12	29
gripper	6/8	6	8	403888	406411	71	3
logistics	16/19	19	16	14265	44111	20	0
mystery	13/13	13	13	95800	26684	17	25
pathways	4/4	4	4	43550	31051	5	8
psr	48/50	48	50	14541	3267	3	0
pw-no-tank	16/21	16	21	122455	295857	48	20
pw-tank	9/13	9	13	127383	85165	211	142
rovers	6/6	6	6	796370	652834	122	10
satellite	6/7	7	6	6287	437238	5	13
schedule-strips	23/50	49	24	3932	152	15	713
tpp	6/6	6	6	1008242	157840	108	2
trucks	6/7	7	6	249518	4586761	198	80
zeno-travel	9/11	9	11	6658	36030	9	1
total	232/292	273	251	107701	457336	44	101